Micriμm

Empowering Embedded Systems

μC/OS-II

μC/Probe

μC/GUI

and the

Atmel AT91SAM9xxx Processors

AT91SAM9261
AT91SAM9263
AT91SAM9RL/R64

Application Note

AN-5002

About Micriμm

Micriμm provides high-quality embedded software components in the industry by way of engineer-friendly source code, unsurpassed documentation, and customer support. The company's world-renowned real-time operating system, the Micriμm **μC/OS-II**, features the highest-quality source code available for today's embedded market. Micriμm delivers to the embedded marketplace a full portfolio of embedded software components that complement **μC/OS-II**. A TCP/IP stack, USB stack, CAN stack, File System (FS), Graphical User Interface (GUI), as well as many other high quality embedded components. Micriμm's products consistently shorten time-to-market throughout all product development cycles. For additional information on Micriμm, please visit www.micrium.com.

About μC/OS-II

Thank you for your interest in **μC/OS-II**. **μC/OS-II** is a preemptive, real-time, multitasking kernel. **μC/OS-II** has been ported to over 45 different CPU architectures and now, has been ported to the Atmel AT91SAM9260, AT91SAM9261, AT91SAM9263, AT91SAM9XE and AT91SAM9RL64/R64 processors

μC/OS-II is small yet provides all the services you would expect from an RTOS: task management, time and timer management, semaphore and mutex, message mailboxes and queues, event flags a much more.

You will find that **μC/OS-II** delivers on all your expectations and you will be pleased by its ease of use.

Licensing

μC/OS-II is provided in source form for **FREE** evaluation, for educational use or for peaceful research. If you plan on using **μC/OS-II** in a commercial product you need to contact Micriμm to properly license its use in your product. We provide ALL the source code with this application note for your convenience and to help you experience **μC/OS-II**. The fact that the source is provided **DOES NOT** mean that you can use it without paying a licensing fee. Please help us continue to provide the Embedded community with the finest software available. Your honesty is greatly appreciated.

About μC/Probe

μC/Probe is a Windows application that allows a user to display the value (at run-time) of virtually any variable or memory location on a connected embedded target. The user simply populates **μC/Probe**'s graphical environment with gauges, tables, graphs, and other components, and associates each of these with a variable or memory location. Once the application is loaded onto the target, the user can begin **μC/Probe**'s data collection, which will update the screen with variable values fetched from the target.

μC/Probe retrieves the values of global variables from a connected embedded target and displays the values in an engineer-friendly format. The supported data-types are: booleans, integers, floats and ASCII strings.

μC/Probe can have any number of 'data screens' where these variables are displayed. This allows to logically group different 'views' into a product.

A 30-day trial version of **μC/Probe** is available on the Micriμm website:

http://www.micrium.com/products/probe/probe.html

Manual Version

If you find any errors in this document, please inform us and we will make the appropriate corrections for future releases.

Version	Date	By	Description
V.1.00	2009/04/03	FT	Initial version.

Software Versions

This document may or may not have been downloaded as part of an executable file, *Micrium-Atmel-uCOS-II–GUI-AT91SAM9xxx.exe*, containing the code and projects described here. If so, then the versions of the Micriμm software modules in the table below would be included. In either case, the software port described in this document uses the module versions in the table below

Module	Version	Comment
μC/OS-II	V2.88	
μC/Probe	V2.3	
μC/GUI	V4.18	

See Also

In addition to the **μC/GUI** projects accompanying this application note, **μC/TCP-IP**, and **μC/Host** and **μC/USB-Device** can be found on the Atmel AT91SAM9 page on the Micrium website.

Document Conventions

Numbers and Number Bases

- Hexadecimal numbers are preceded by the "0x" prefix and displayed in a monospaced font. Example: `0xFF886633`.
- Binary numbers are followed by the suffix "b"; for longer numbers, groups of four digits are separated with a space. These are also displayed in a monospaced font. Example: `0101 1010 0011 1100b`.
- Other numbers in the document are decimal. These are displayed in the proportional font prevailing where the number is used.

Typographical Conventions

- Hexadecimal and binary numbers are displayed in a monospaced font.
- Code excerpts, variable names, and function names are displayed in a monospaced font. Functions names are always followed by empty parentheses (e.g., `OS_Start()`). Array names are always followed by empty square brackets (e.g., `BSP_Vector_Array[]`).
- File and directory names are always displayed in an italicized serif font. Example: */Micrium/Sofware/uCOS-II/Source/*.
- A bold style may be layered on any of the preceding conventions—or in ordinary text—to more strongly emphasize a particular detail.
- Any other text is displayed in a sans-serif font.

Table of Contents

1. Installing the Micirum Software 7

2. Using the Micirum IAR Example Project 8
2.01.01 IAR **μC/OS-II** Kernel Awareness. 9
2.02 Running the Example Applications 9
2.02 IAR Project configuration 12
2.03 IAR AT91SAM9xxx boot loaders 12

3. μC/Probe 13

Licensing 16
References 16
Contacts 16

1. Installing the Micirum Software

A compiled binary example projects are provided in the self-extracting executable the self-extracting executable ***Micrium-Atmel-uCOS-II-GUI-AT91SAM9xxx.exe.***

You will be prompted to accept the simple terms of the licensing agreement. If you answer 'Yes', the software will be installed on your PC under the ***\Micrium*** directory from the root as shown in Figure 1-1

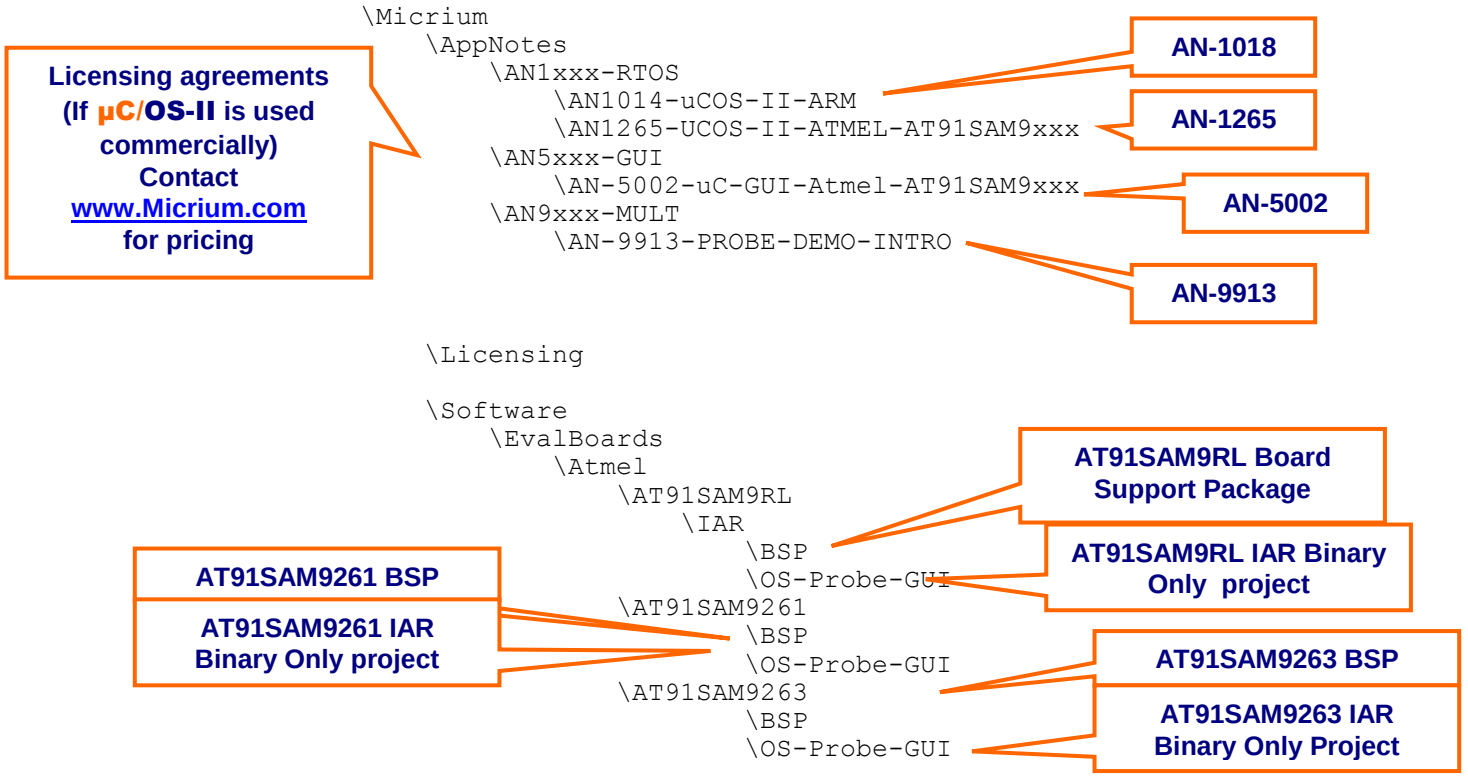

Figure 1-1. Directory Structure

2. Using the Micirum IAR Example Project

To view the example project, start an instance of IAR Embedded Workbench, and open:

- For the Atmel AT91SAM9RL-EK evaluation board , ***AT91SAM9RL--OS-Probe GUI-BIN.ewp***, located in
  /Micrium/Software/EvalBoards/Atmel/AT91SAM9RL-EK/IAR/OS-Probe-GUI

- For the Atmel AT91SAM9261-EK evaluation board , ***AT91SAM9261--OS-Probe GUI-BIN.ewp***, located in
  /Micrium/Software/EvalBoards/Atmel/AT91SAM9261-EK/IAR/OS-Probe-GUI

- For the Atmel AT91SAM9263-EK evaluation board , ***AT91SAM9263--OS-Probe GUI-BIN.ewp***, located in
  /Micrium/Software/EvalBoards/Atmel/AT91SAM9263-EK/IAR/OS-Probe-GUI

To do this, use the *Add Existing Project...* menu command under the *Project* menu:

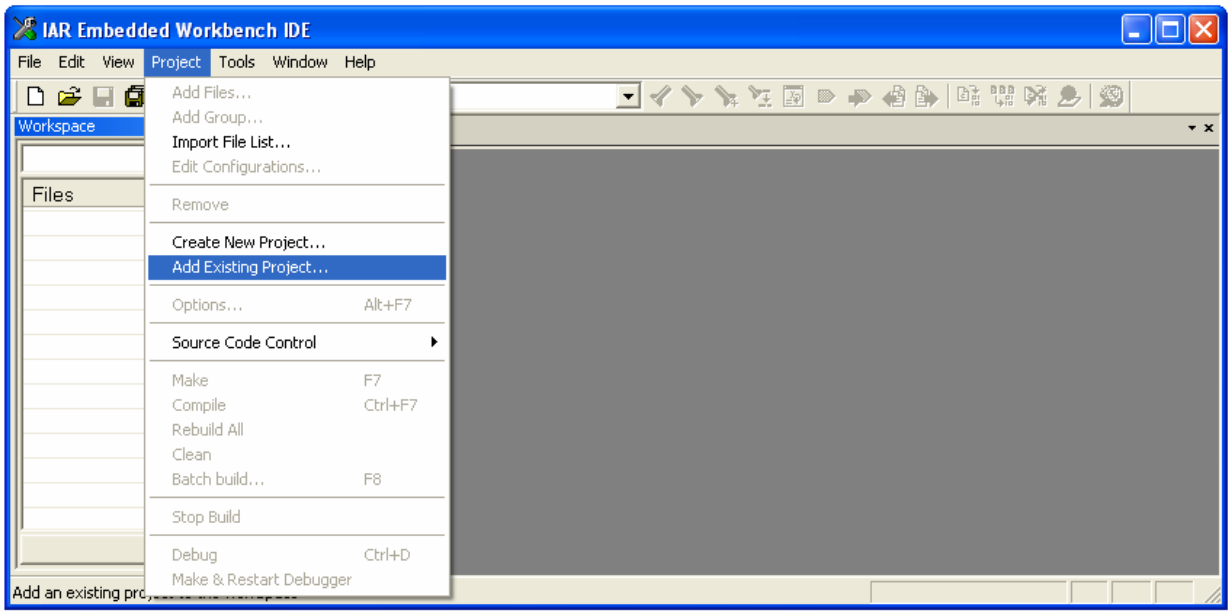

Figure 2-1. IAR EW. Opening an existing project

IAR EWARM

(1) Be certain to open the proper project for your version of EWARM. The Atmel AT91SAM9xxx examples projects were built using **EWARM ver. 5.3**

(2) You might receive warning messages while running the debugger, this is because the source code is not provided and IAR con located the files. In this case the message should be ignored.

2.01.01 IAR μC/OS-II Kernel Awareness.

The **μC/OS-II** Kernel Awareness plug-in will allow you to examine information about system objects while using the C-Spy debugger. To gain access to this feature, enable the plug-in by right-clicking on the project name in the work space browser and choosing *Options…* Then, select the "Debugger" entry in the list box, and the "Plug-in" tab pane. Find the **μC/OS-II** entry in the list and, finally, select the check box beside the entry.

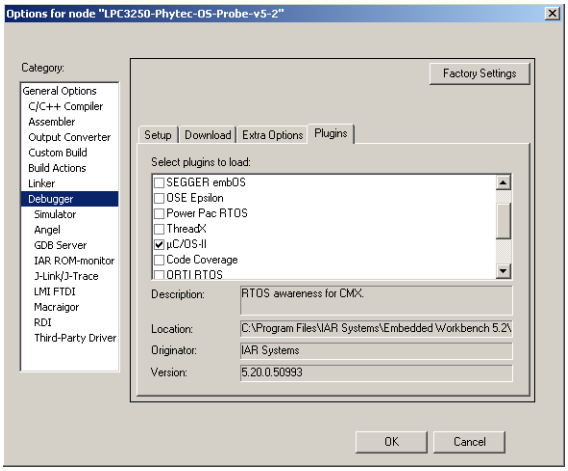

Figure 2-2. Enabling the μC/OS-II Kernel Awareness Plug-In

2.02 Running the Example Applications

The example project includes a basic demonstration of **μC/OS-II** and **μC/Probe.** The evaluation board components are labeled in the figures 2-3 to 2-5:

The example application is the same for all AT91SAM9xxx-EK boards

To communicate with the board through RS-232, connect a serial cable between the evaluation board serial port "DBG port" and your PC and open a HyperTerminal or any Terminal program window. Configure the RS-232 interface with the following settings:

- **Bits per Second**: 115200
- **Data bits**: 8
- **Parity:** None
- **Stops bits:** 1
- **Flow Control:** None

Once the program is loaded onto the target, the LEDs will start blinking.

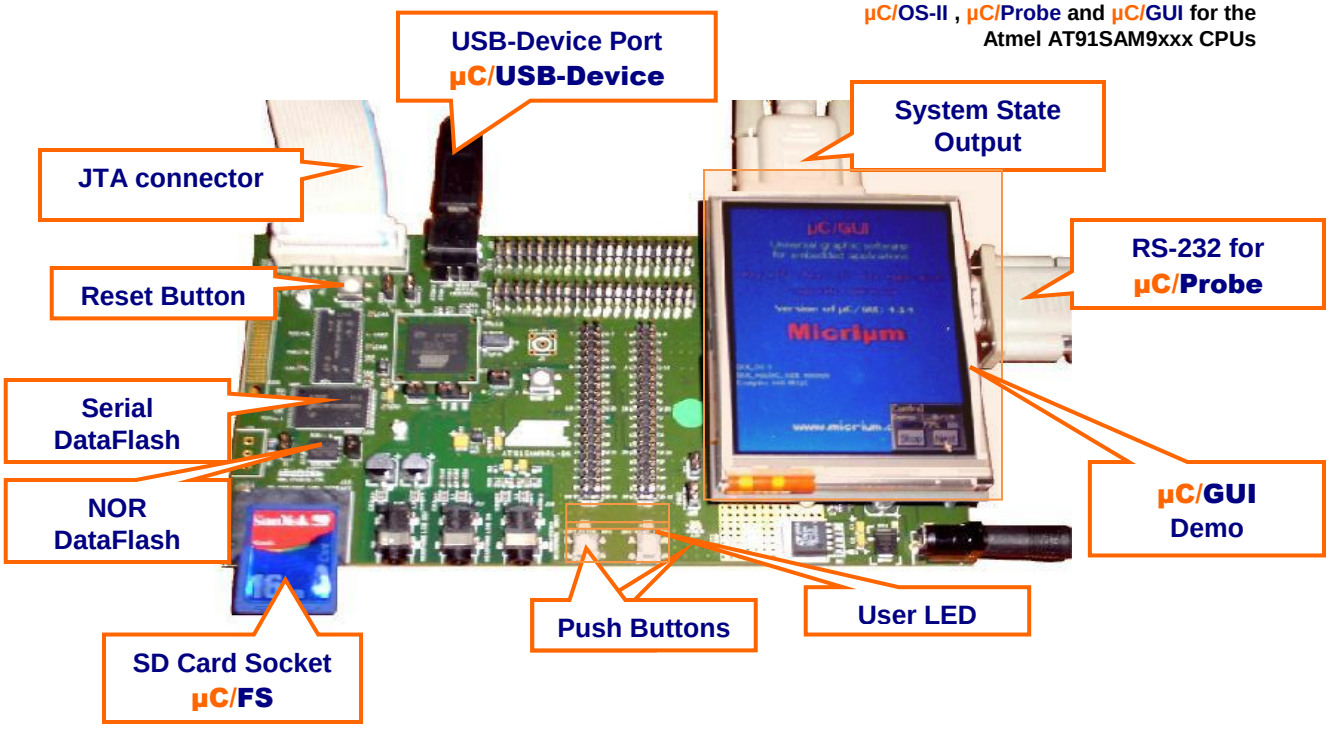

Figure 2-3. Atmel AT91SAM9RL-EK Development Board

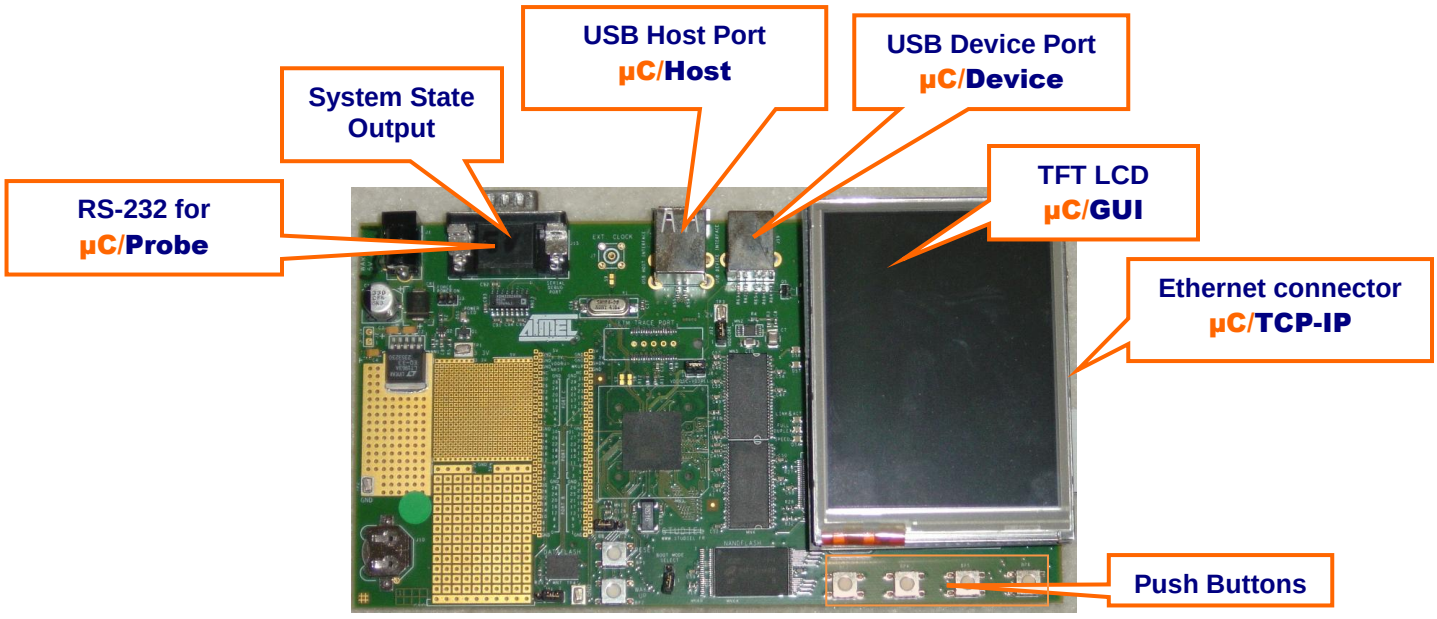

Figure 2-4. Atmel AT91SAM9261-EK Development Board

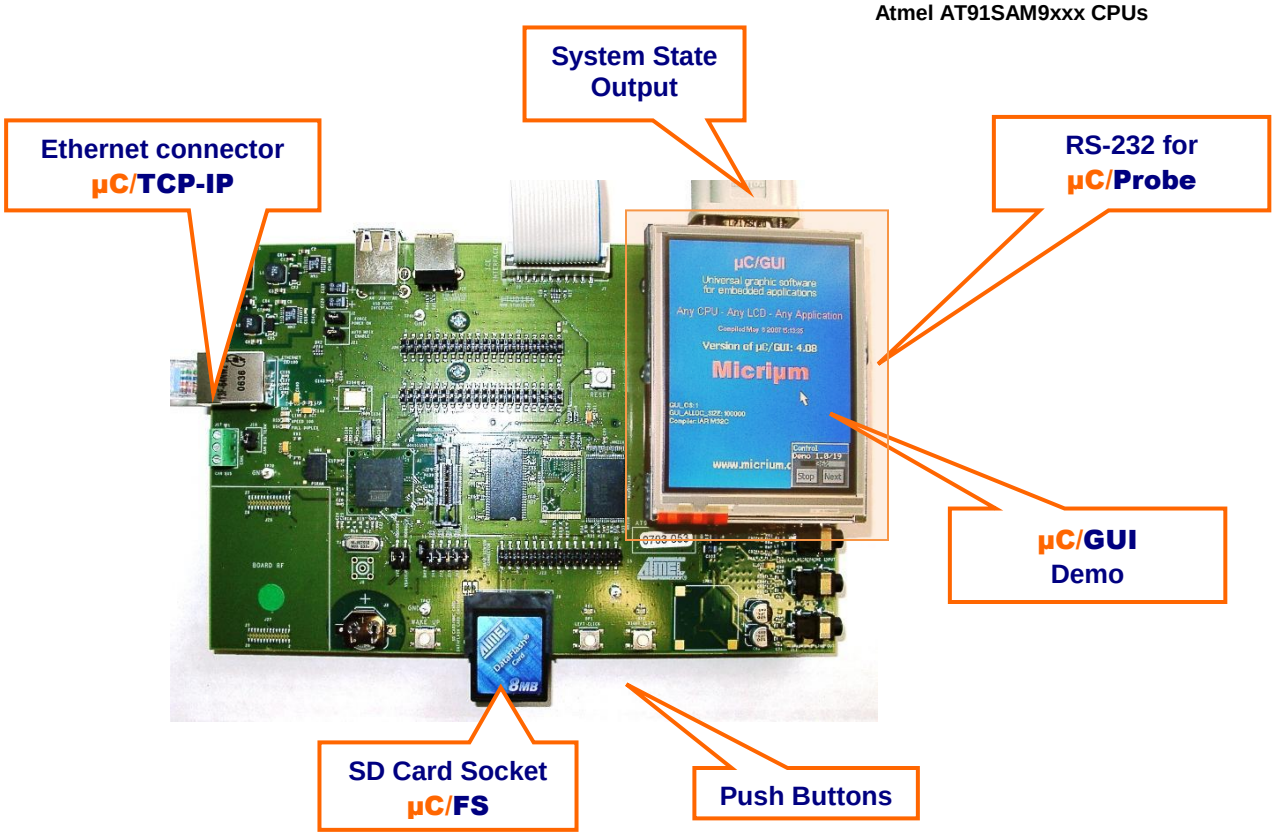

Figure 2-5. Atmel AT91SAM9263-EK Development Board

The RS232 port labeled “RS232 for **μC/Probe**” is used for **μC/Probe** (at 115200 baud), which allows you to view (in real-time) the value of any variables in the target system.

Figure 2-6 show an example of the HyperTerminal window.

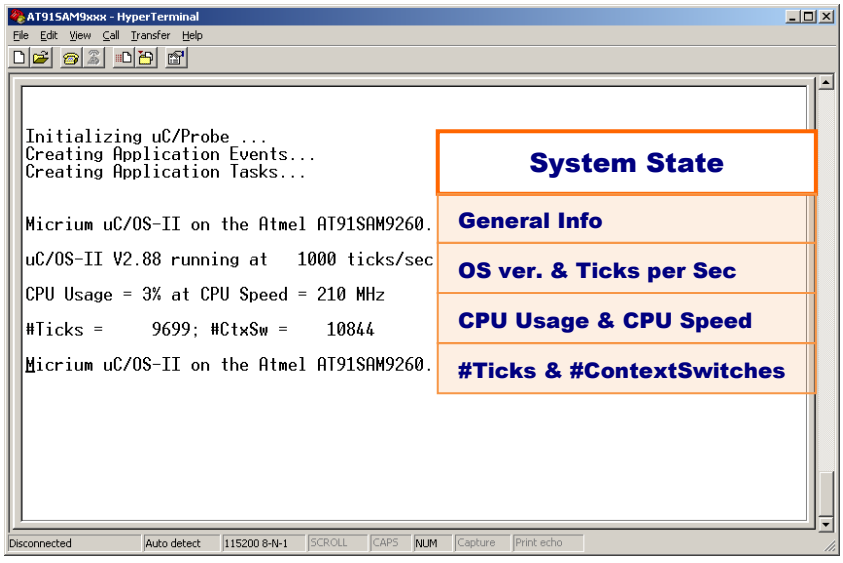

Figure 2-6. System State

The push buttons can be used to change the output to a new item.

The **μC/GUI** demo should be displayed in the AT91SAM9xxx-EK board's LCD display as shown in Figure 2-3 and 2-5.

2.02 IAR Project configuration

The IAR project configurations allow you to compile, link and load the software in different ways to the target. The following configuration is available in the IAR projects.

- XRAM-BIN: This project option is configured to load the code into the board's external SDRAM (XRAM)

2.03 IAR AT91SAM9xxx boot loaders

The example applications can be loaded onto the processor via a J-TAG, as described in Section 2.02. Because some of the projects use volatile memory (XRAM, IRAM) the code will not be retained when the processor is reaset or power is interrupted.

In order to boot from the non-valtile memory a boot loader is needed. IAR provides examples that allow allow to program internal flash memories, as well as external flash memories such as Nandflash, Dataflash, Serialflash, Norflash or TWI EEPROM.

The document flashloaders.html explains all the details to integrate the boot loader in your application. This document can be found in the IAR embedded workbench installation folder:

/Program Files/IAR Systems/Embedded Workbench xxx/arm/examples/Atmel/ flashloaders.html

Where 'xxx' is the current version of the IAR EWARM.

In addition to the IAR bootloaders, there is also a bootstrap (AT91Bootstrap) application for AT91SAM9 microcontrollers. It can be uploaded and launched by the ROM Boot Program and can be used to configure the system and to download a larger application. For more information vist the Atmel's web site.

3. μC/Probe

μC/Probe is a Windows program which retrieves the values of global variables from a connected embedded target and displays the values in a engineer-friendly format. To accomplish this, an ELF file, created by the user's compiler and containing the names and addresses of all the global symbols on the target, is monitored by **μC/Probe**. The user places components (such as gauges, labels, and charts) into a Data Screen in a **μC/Probe** workspace and assigns each one of these a variable from the Symbol Browser, which lists all symbols from the ELF file. The symbols associated with components placed on an open Data Screen will be updated after the user presses the start button (assuming the user's PC is connected to the target).

μC/Probe currently interfaces with a target processor with a RS-232. A small section of code resident on the target receives commands from the Windows application and responds to those commands. The commands ask for a certain number of bytes located at a certain address, for example, "Send 16 bytes beginning at 0x0040102C". The Windows application, upon receiving the response, updates the appropriate component(s) on the screens with the new values.

Start Button.
This button switches between Design and Run-Time Views. During Run-Time View (when data is collected), this will appear as a stop button (a blue square).
Symbol Browser.
Contains all symbols from the ELF files added to the workspace.
Data Screen.
Components are placed onto the data screen and assigned symbols during Design View. During Run-Time View, these components are updated with values of those symbols from the target
Callback Counter
A counter is kept in the uC/Probe OS plug-in callback, which is displayed with this gauge.
Atmel AT91SAM9263-EK
Push Buttons
LEFT CLICK
RIGHT CLICK
Serial Output
Micrium uC/OS-II on the Atmel AT91SAM9263
www.micrium.com

Figure 6-1. μC/Probe Windows Program

To use **μC/Probe** with the example project (or your application), do the following:

1. **Download and Install μC/Probe.** A trial version of **μC/Probe** can be downloaded from the Micriμm website at

   http://www.micrium.com/products/probe/probe.html

2. **Open μC/Probe**. After downloading and installing this program, open the example **μC/Probe** workspace for **μC/OS-II**, named ***OS-Probe-Workspace.wsp***, which should be located in your installation directory at

   /Program Files//Micrium/uC-Probe/Target/Plugins/uCOS-II/Workspace

3. **Connect Target to PC**. Currently, **μC/Probe** can use RS-232 to retrieve information from the target. You should connect a RS-232 cable between your target and computer.

4. **Load Your ELF File**. The example projects included with this application note are already configured to output an ELF file. (If you are using your own project, please refer to Appendix A of the **μC/Probe** user manual for directions for generating an ELF file with your compiler.) This file should be in

   /<Project Directory>/<Configuration Name>/exe/

   where <***Project Directory***> is the directory in which the IAR EWARM project is located (extension *.ewp) and <***Configuration Name***> is the name of the configuration in that project which was built to generate the ELF file and which will be loaded onto the target. The ELF file will be named

   <Project Name>.elf

   in EWARM v4.4x and

   <Project Name>.out

   in EWARM v5.1x unless you specify otherwise. To load this ELF file, right-click on the symbol browser and choose “Add Symbols”.

5. **Configure the RS-232 Options**. In **μC/Probe**, choose the “Options” menu item on the “Tools” menu. A dialog box as shown in Figure 6-2 (left) should appear. Choose the “RS-232” radio button. Next, select the “RS-232” item in the options tree, and choose the appropriate COM port and baud rate. The baud rate for the projects accompanying this appnote is 115200.

6. **Start Running**. You should now be ready to run **μC/Probe**. Just press the run button ( ) to see the variables in the open data screens update. Figure 6-3 displays two screens in the **μC/OS-II** workspace which display detailed information about each task’s state.

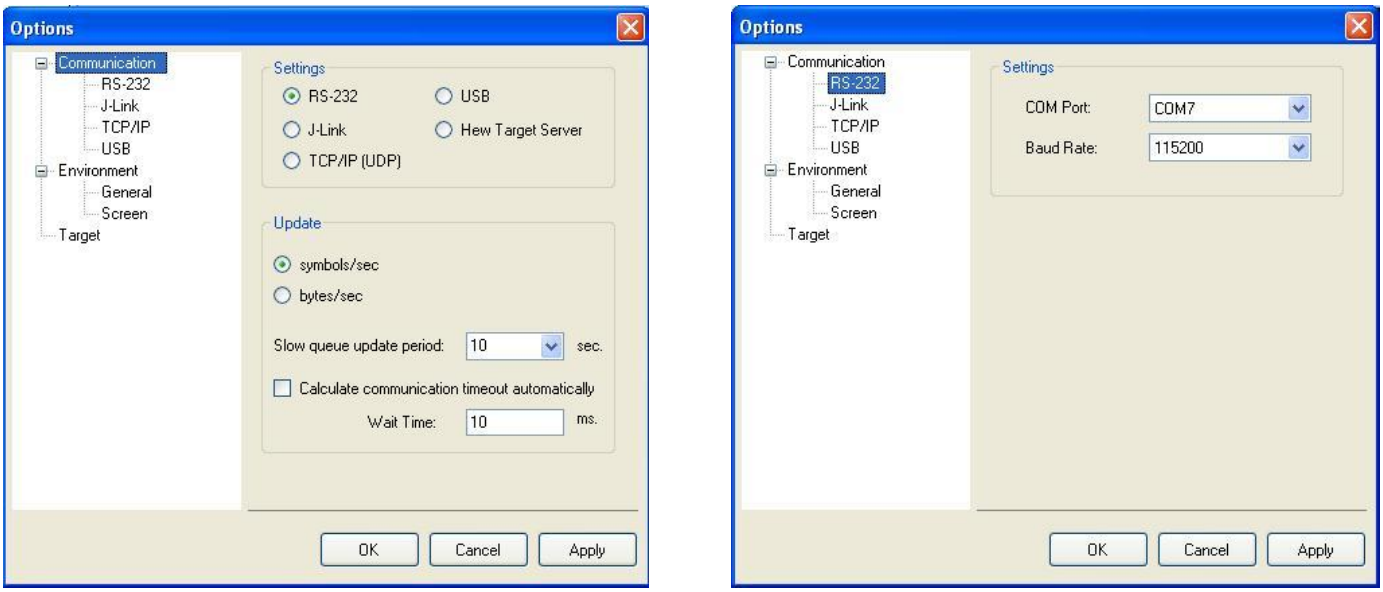

Figure 6.2. μC/Probe Options

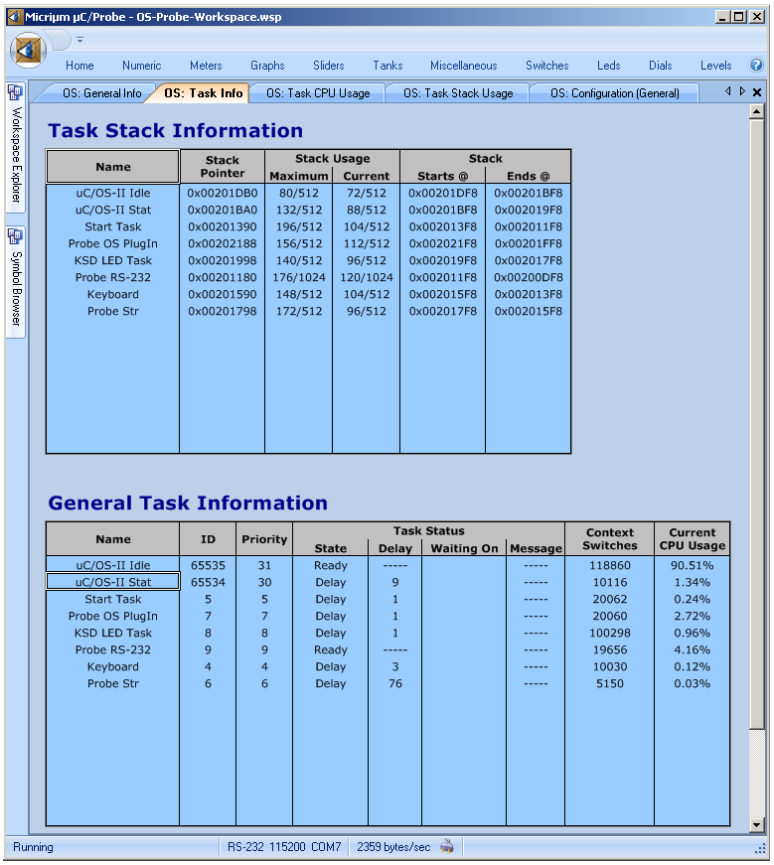

Figure 6-3. μC/Probe Run-Time: μC/OS-II Task Information

Licensing

μC/OS-II is provided in source form for **FREE** evaluation, for educational use or for peaceful research. If you plan on using **μC/OS-II** in a commercial product you need to contact Micriμm to properly license its use in your product. We provide **ALL** the source code with this application note for your convenience and to help you experience **μC/OS-II**. The fact that the source is provided does **NOT** mean that you can use it without paying a licensing fee. Please help us continue to provide the Embedded community with the finest software available. Your honesty is greatly appreciated.

References

μC/OS-II, The Real-Time Kernel, 2nd Edition
Jean J. Labrosse
R&D Technical Books, 2002
ISBN 1-57820-103-9

Embedded Systems Building Blocks
Jean J. Labrosse
R&D Technical Books, 2000
ISBN 0-87930-604-1

Contacts

IAR Systems
Century Plaza
1065 E. Hillsdale Blvd
Foster City, CA 94404
USA
+1 650 287 4250
+1 650 287 4253 (FAX)
e-mail: Info@IAR.com
WEB : www.IAR.com

CMP Books, Inc.
1601 W. 23rd St., Suite 200
Lawrence, KS 66046-9950
USA
+1 785 841 1631
+1 785 841 2624 (FAX)
e-mail: rushorders@cmpbooks.com
WEB : http://www.cmpbooks.com

Micriμm
949 Crestview Circle
Weston, FL 33327
USA
+1 954 217 2036
+1 954 217 2037 (FAX)
e-mail: Jean.Labrosse@Micrium.com
WEB : www.Micrium.com

Atmel
2325 Orchard Parkway
San Jose, CA 95131
USA
+1 408 441 0311
+1 408 487 2500
WEB : www.atmel.com